Timo Stich,
Christian Linz,
Christian Wallraven,
Douglas Cunningham,
Marcus Magnor

Perception-based Image Interpolation

Technical Report / Computer Graphics Lab, TU Braunschweig ; 2008-2-5

Veröffentlicht: 06.02.2008

http://www.digibib.tu-bs.de/?docid=00021922

TIMO STICH, CHRISTIAN LINZ, CHRISTIAN WALLRAVEN,
DOUGLAS CUNNINGHAM, MARCUS MAGNOR

Perception-based Image Interpolation

Technical Report 2008-2-5

February 5, 2008

Computer Graphics Lab, TU Braunschweig

Chapter 1

Abstract

We present a perception-based method for image interpolation, aiming for perceptually convincing transitions between real-world images. Without 3D geometry or scene motion, perception-based image interpolation enables smooth viewpoint navigation across space and time. We show how global visual effects can be created from a collection of unsynchronized, uncalibrated images. A user study confirms the perceptual quality of the proposed image interpolation approach.

Chapter 2

Introduction

Frozen moment, slow motion, live action: many stunning global visual effects are known in the F/X community [Wol06]. To capture such visual effects directly from real-world dynamic scenes, as is typically done today [TM08], requires sophisticated acquisition hardware, meticulous shot planning, and elaborate recording setup procedures. While such images can be easily rendered in digitally modeled dynamic scenes using standard computer graphics techniques, modeling *real-world scenes* is at least as time-consuming and tedious as capturing the effect images directly [VBK05], with no guarantee of obtaining photo-realistic rendering results. There is, however, another alternative: one could *interpolate* the effect images directly from conventionally acquired footage.

Chen and Williams were among the first to propose that one interpolate intermediate viewpoints from a set of real-world photographs[CW93]. Their method, like all subsequent image-based rendering (IBR) techniques, ensured perceptionally plausible results by enforcing physically appropriate constraints. This requires camera calibration parameters and, frequently, scene geometry.Time-varying scenes further require that the cameras be synchronized, so that images of the same time instant can be compared.Clearly, the need for calibrated, synchronized acquisition is highly inconvenient as it implies time-consuming recording preparations as well as expensive acquisition hardware. Instead, a generally applicable image interpolation approach yielding perceptually authentic results would be able to interpolate across space and time from nothing more than a collection of unsynchronized, uncalibrated images.

It is well-known that our brain automatically interprets changing visual input in terms of plausible motion of the viewpoint and/or of the observed scene [Wer38, Gra65, GP00, GP03]. Critically, the brain apparently does *not* rely on the laws of physics for its judgements [BN92, SD96, Wol98]. Cartoon animations, for example, frequently contain motion which, while not physically accurate, is perceptually quite plausible. Here, we therefore propose a perceptual approach to image inter-

polation, which takes visual motion perception into consideration to create smooth transitions between still images that our brain accepts as plausible motion. A user study confirms the perceptual validity of the proposed approach. By eliminating the need for camera calibration and synchronization, perception-based image interpolation significantly simplifies data acquisition for image- and video-based rendering applications. Furthermore, our approach enables authoring various different visual effects from the same input images, as well as optimizing parameter settings, during post-processing. Finally, completely new effects become possible, e.g., by applying scene-adaptive motion compensation during interpolation.

Chapter 3

Related Work

Perceptual Graphics is a steadily growing interdisciplinary research field which seeks to integrate perceptual psychology and computer graphics[OHM+04]. Recent topics include human perception of global illumination effects[MTAS01], the influence of shape on material perception[VLD07], and visual equivalence of rendered images [RFWB07].

Image metamorphosis, or image morphing, denotes interpolation between images depicting different objects from user-defined correspondences. One of the most well-known examples is line-based morphing proposed by Beier and Neely [BN92], used in Michael Jackson's music video "Black & White". Other warping techniques have been discussed by Wolberg [Wol98], including the popular thin-plate spline interpolation which is based on point correspondences. A computationally more complex method based on line features was recently proposed by Schaefer et al. [SMW06]. In general, image morphing is based on a continuous 2D vector field denoting dense image correspondences along which both images are warped and linearly blended to obtain in-between images. Such a simplistic motion model, however, cannot properly handle dynamic occlusions or motion discontinuities.

Optical flow refers to the flow field created by the spatiotemporal trajectories of image patches during an image sequence, and was first described by the psychologist James J. Gibson [Gib55]. Since the pioneering work on local and global optical flow reconstruction by Lucas and Kanade [LK81] and Horn and Schunck [HS81], respectively, a multitude of computational approaches have been devised and applied in a variety of fields [BFB94, BM04]. Optical flow is frequently used to represent the motion field between two images, even though optical flow approaches cannot account for occlusions or disocclusions. For small motions, a solution to this problem was recently proposed by [LTF+05] by segmenting the images into separate motion layers.

Image-based rendering (IBR) methods achieve highly realistic rendering results using a collection of calibrated photographs. While some IBR methods rely solely on the number of images to minimize aliasing artifacts [LH96, MP04], most IBR approaches make additional use of epipolar constraints [MB95, SD96, MBR+00], scene depth [CW93, GGSC96, IMG00, BBM+01, ZKU+04], or full 3D geometry information [DBY98, WAA+00, CTMS03, SSS06]. To create visual effects as described by Wolf or Taylor [Wol06, TM08], only spatiotemporal view interpolation [VBK05] is able to recover the scene flow over time [VBR+05]. The crucial disadvantage of IBR techniques is the need for accurate camera calibration, additional geometry modeling, and/or synchronized acquisition. These limitations make data acquisition for IBR a time-consuming and delicate endeavour which typically requires a controlled environment and expensive equipment.

In the next chapter, we describe how dense image correspondences can be established taking visual motion perception into account. Sect. 5 presents the perception-based interpolation algorithm. Chapter 6 presents a user study evaluating our approach. Results are presented in Sect. 7. We conclude with a discussion of our contributions and an outlook on promising future research directions.

Chapter 4

Perceptual Image Correspondences

Interpolating between two images is straight-forward if the true correspondence field is known. Robustly establishing the true correspondence field from the images alone, however, is a formidable task that, in general, cannot be solved. Fortunately, we do not need the true correspondence field if our goal is to generate *perceptually convincing* interpolation results.

4.1 Visual Motion Perception

Human vision is a very powerful system, adept at extracting meaningful patterns so that we can understand, navigate through, and interact with our surroundings rapidly and efficiently. The importance and complexity of this task is perhaps reflected by the fact that approximately half of our brain is dedicated to processing visual input. The earliest stages of vision, also called low- and intermediate-level vision, are comparatively well understood, particularly motion perception. Based on his work with flies and beetles, [Rei61] mathematically and neurally described a local-correlator motion detector. The detector, which explicitly relies on the fact that real-world objects tend to move rather smoothly, matches small image patches across small spatial and temporal distances. Interestingly, low-level motion processing in humans is also well described by this detector [QA97, HBD$^+$99]. Additionally, the human visual system seems to take advantage of the fact that neighboring areas on an object tend to have the same motion. This allows local smoothness constraints to help compensate for noise and aids in image segmentation. Finally, the common motion of neighboring patches and differently oriented edges are used to help solve the aperture problem [Wal35] . To achieve perceptually plausible image interpolation results, then, it is important to transform edges exactly and

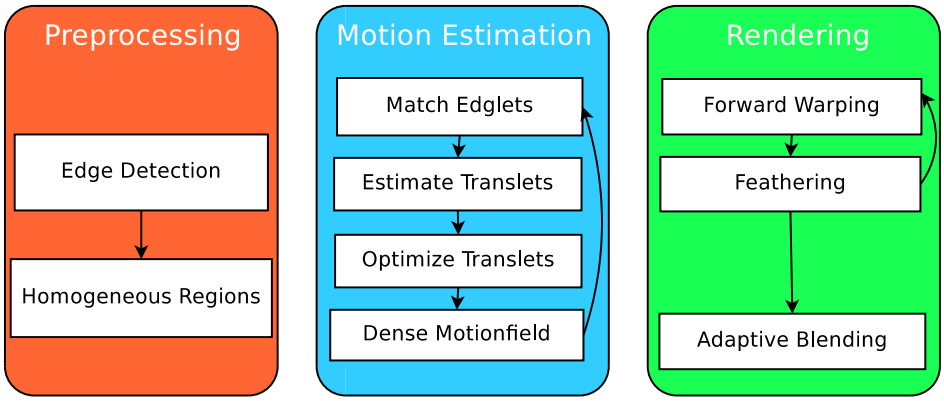

Figure 4.1: *Overview of perception-based image interpolation: first, the images are preprocessed to find edges and homogeneous regions. These are then used to determine a perceptually plausible correspondence field. Finally, we use this correspondence field for interpolation rendering of image transitions in real-time.*

homogeneous regions coherently. Thus, the critical aspects of an image sequence that will result in perceptually plausible motion: edges, homogeneous regions, and coherent motion.

4.2 Algorithm Overview

Based on the perceptual criteria outlined above, Figure 4.1 gives an overview of the proposed interpolation approach. It is composed of three separate parts. First, the images are preprocessed to find edges and homogeneous regions. Second, a global transformation between the images is computed, which we achieve by first matching edges between the images. The global correspondences are then based on estimated local transformations of homogeneous image regions using these matches. In general, this algorithm is iterated three to four times until it converges. Third, we use these corresondences to interpolate between the images. The rendering is implemented on standard graphics hardware and runs at real-time frame rates.

4.3 Edges

Edges convey a great deal of information about the image in a very compact way, and the human visual system relies heavily on them. Hildreth and Marr proposed an algorithm to find edges that is motivated by the physiological model of so called simple cells [MH80]. We use a more recently proposed edge detector, the Compass operator [RT99]. Critically for our purposes, the edges found by this operator closely resemble the ones that a human observer would highlight. Thresholding

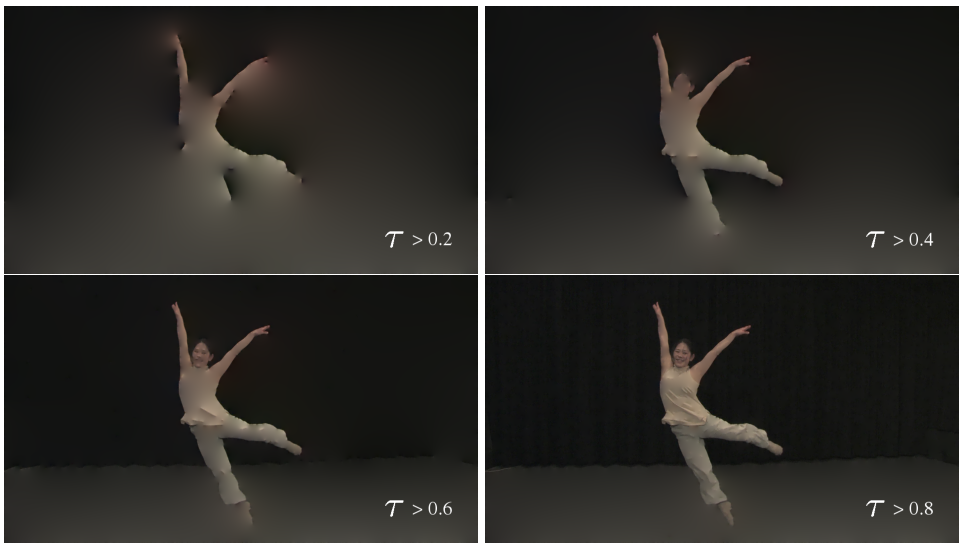

Figure 4.2: *In this series, the original image is approximated by its edglets using different thresholds on the edglet strength τ. Non-edglet regions are filled in by isotropic diffusion. Note how we quickly grasp the gist of the image contents even though less than one percent of the original image information is used for all reconstructions.*

with the strength of the operator on the found edges τ results in a set of edge pixels e_i, called *edglets*, that represent the perceptually relevant edges in the image. As can be seen in Figure 4.2, the approximation of the image by only the color information around the edglets quickly converges towards the original image in the sense of perceived difference. We found that for natural images, between 2000 to 20000 edglets, i.e. edge pixels, are sufficient to represent the edges that are critical for perceived motion.

4.4 Homogeneous Regions

Natural images can be seen as composites of connected homogeneous image regions. Indeed, the Gestalt psychologists demonstrated the central role that grouping image elements plays in image understanding [Wer38]. Consistent with this observation, [RM03] proposed that connected pixels with similar properties can be grouped into superpixels. These can then be used as the building blocks for a higher order decomposition. [FH04] proposed a very interesting, perceptually-based algorithm for segmenting images. The method outlined in the following is based on this approach, and produces high quality, perceptually meaningful superpixels for natural images very quickly. Our approach first combines the result of the edge detector with the results of the standard superpixel decomposition, Fig-

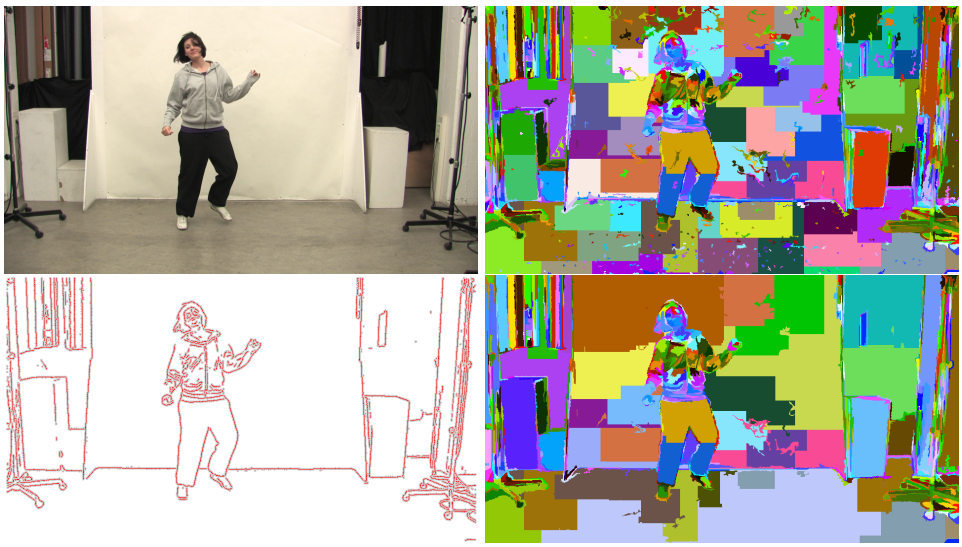

Figure 4.3: *An image (upper left) and its decomposition into its homogeneous regions (upper right). Since the transformation estimation is based on the matched edglets, only superpixels that contain actual edglets (lower left) are of interest. We merge superpixels with insufficient edglets with their neighbors (lower right).*

ure 4.3. Since the transformation estimation is based on the matched edglets, only segments that contain actual edglets are of interest. Thus, we merge superpixels with an insufficient number of edglets (4 in our case) with their spatial neighbors. This results in a further optimized over segmentation of the image based only on the image itself. Note that since we only use this decomposition as an initial partitioning of the image, the results must not necessarily be exact in the sense that only meaningful parts of real world objects should correspond to a single segment. The only requirement is that the combined spatial support of the non-overlapping segments m_i spans the whole image and that all motion discontinuity borders in the correspondence field can be described as borders of this decomposition.

4.5 Matching Edglets

After the images are preprocessed, we can compute a perceptually plausible global transformation between the images. This is achieved by iterating the following steps (Sections 4.5 to 4.8) until convergence.

The driving force for the global transformation are matches between the edglets. There are many possible approaches to solve such a feature matching problem, the simplest of which is to rely only on the Euclidean distances between the features. Since this is, however, often not sufficient to obtain a good solution, additional

descriptors need to be computed. A large part of these descriptors is computed from the local image structure around each feature.

Popular local descriptors such as the SIFT features [Low04], however, become unreliable when matching over non-rigid deformations since they are too sensitive to changes in pixel values around the features. When interpolating between, for example, two time steps of an image sequence, these regions often change significantly and the matching is bound to fail. Other descriptors are based on the spatial distribution of the features themselves, such as for example the Shape Context of [BMP01]. These are often more robust since they are based on more abstract version of the images to match. We use a localized version of the Shape Contexts.

Similar to [BMP01], we regard the matching of the edglets as an optimal assignment problem encoded in a weighted graph, where each edglet e_i in the first edglet set is either matched to an edglet e'_j in the second image or to a virtual edglet v_i. We introduce virtual edglets for each source edglet since we want to account for occluded edglets while still computing a complete assignment. Our cost for matching a pair of edglets in iteration k is defined as

$$C^k(e_i, e'_j) = D(||H^{k-1} \cdot e_i - e'_j||_2) \cdot ||S_i - S'_j||_2 \tag{4.1}$$

where S_i, S'_j are the corresponding Shape context to the edglets e_i and e'_j, and H^{k-1} is the identity transformation or a previously computed transformation matrix for edglet e_i in iteration $k > 0$ (cf. Section 4.6). The transformation $H^{k-1}_{e_i}$ results from the previous iteration and is used to get an improved starting point for the matching in this iteration. The distance measure D is defined as

$$D(x) = \frac{a}{(1 + e^{-bx})} \tag{4.2}$$

with $a, b > 0$ such that the maximal cost for the Euclidean distance is limited by a. Each edglet from the first image is also connected to a virtual edglet v_i. The user-defined cost $C^k(e_i, v_i)$ controls how aggressively the algorithm tries to find a match with a real edglet before matching with the virtual edglet, which is equivalent to classifying the edglet as occluded in the second image. In each iteration, the reduction of this cost avoids more mismatches as only the best matches up to this threshold are found. This global minimal cost assignment problem is solved by applying the auction algorithm [Ber92]. This algorithm proved to be the fastest global optimal algorithm for our assignment problem.

Although the proposed matching algorithm works sufficiently well, in complex situations with a lot of occlusion, wrong parts are sometimes matched. In this case the user can manually correct the mismatch by coarsely marking the corresponding parts of the images. By restricting the set of possible matches the correct match is then again automatically found and can be used to correct the solution.

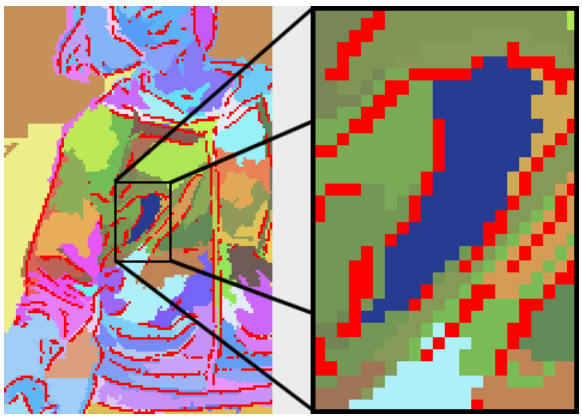

Figure 4.4: *A translet is defined as the pair of an image segment (green) and the subset of edglets that are within the spatial support of this segment (red). For each translet a transformation is estimated from the computed matches of its edglets.*

4.6 Translet Estimation

After we have computed a match between the edglets, we use this information to find a global image transformation also for non-edglet pixels. A *translet* t is defined as the pair of an image segment s, computed by the preprocessing described in Chapter 4.4, and the subset of edglets $e_i : e_i \in s$ that are located inside the spatial support of this segment, Figure 4.4. For each translet a transformation is estimated from the previously found matches of its edglets, e.g. a similarity, affine or perspective transformation. Although one of these simple transformations can hardly represent the motion between images in general, the combination of hundreds of such local transformations performs very well. Each translet has between 4 to 100 edglets depending on the spatial support. The robustness of the estimation is thereby further increased by filtering outliers using the RANSAC algorithm [HZ00]. Note that edglets of a translet need not be part of a corresponding translet in the second image. We use them solely for partitioning of the image into homogeneous parts

4.7 Translet Optimization

At this point, we have established dense correspondences between the images by finding local transformations that describe the deformation of the matched edglets. In the next step, we further optimize the current solution by again regularizing the local translets. The initial segmentation is generally very conservative, such that the support of the translets is too small for a reliable estimation. The fact that neighboring translets, in general move similarly can be used to optimize the

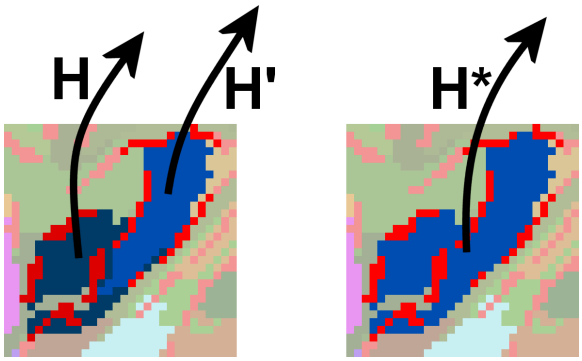

Figure 4.5: *During optimization similar transformed neighboring translets are merged into a single translet. After merging, the resulting translet consists of the combined spatial support of both initial translets (light blue and dark blue) and their edglets (light red and dark red).*

current solution. Using a greedy approach, we iteratively merge the current most similar transformed neighboring translets into one, as depicted in Figure 4.5, until a user-defined threshold is reached. When two translets are merged, the resulting translet then contains both edglet sets and has the combined spatial support. The transformation is re-estimated based on the new edglet set where outliers are again effectively removed using RANSAC filtering.

In the first iteration of the whole estimation process, a large threshold is used to get only a few final translets. In subsequent iterations, the threshold is successively reduced to allow for more and more variation in the correspondence field as the number of outliers in the matching increases.

4.8 Global Transformation Field

Although the current deformation is smooth within the optimized translets, some unwanted discontinuities can still be present. For example, when only a part of a translet boundary is at a true motion discontinuity, noticeably incorrect discontinuities can produce artifacts along the rest of the boundary. This can only be solved on a per pixel basis. The first step in addressing this issue is thus the computation of the corresponding deformation vector for each pixel of the image. Since the translets partition the image, each pixel p_i in the image is uniquely associated with a translet t. Its deformation vector is then computed by

$$d(p_i) = H_t \cdot p_i - p_i. \tag{4.3}$$

We can now apply anisotropic diffusion [PM90] on this vector field using the diffusion equation

$$\delta I/dt = div(\ g(min(|\nabla d|, |\nabla I|)\ \nabla I) \tag{4.4}$$

which is dependent on the image gradient ∇I and the gradient of the deformation vector field ∇d whichever is smaller in magnitude at the observed pixel. The function g is a simple mapping function as defined in [PM90]. Thus, the deformation vector field is smoothed in regions that have similar color or similar deformation, while discontinuities that are both present in the color image and the vector field are preserved. This again improves the overall smoothness of the final result while preserving perceptually important motion discontinuities. During the anisotropic diffusion, we keep the motion vectors of the edglets that are marked as inliers as boundary values to still ensure exact edglet motion.

In total, computing the dense correspondence field takes no more than 60 seconds for images of size 960x540 pixels on a AMD Athlon64 system.

Chapter 5

Interpolation Rendering

Rendering in-between images is achieved by applying the correspondence field to the images using warping and then blending the warped images. This can be done in realtime on graphics hardware using per-vertex mesh deformation and alpha blending. In this chapter, we discuss the issues of missing regions and fold-over at motion discontinuities. We also show how to combine multiple images into a single in-between image using locally varying blending parameters and feathering of unnaturally hard color transitions that can occur at fold-overs. This enables us to interpolate in real-time between several images at once allowing to create, for example, full space-time interpolation in both space and time at the same time.

5.1 Interpolation of the Transformation

We need to interpolate the dense transformation from the identity transformation to the computed transformation to get an in-between transformation. The simplest and fastest approach is to use the precomputed deformation vector field d_{AB} between the images A and B and linearly interpolate the transformation multiplying with the appropriate factor $\alpha \in [0 \dots 1]$

$$d_{AB}(\alpha) = \alpha \, d_{AB} \tag{5.1}$$

This may not, however, yield plausible results for large angle rotations. An alternative would be to decompose the transformations into their components, e.g. translation, rotation and scaling in the affine case, and then separately interpolate similar to the as-rigid-as possible deformation approach by [ACOL00]. This, however, requires the final anisotropic diffusion to be recalculated to get a per-pixel correct correspondence field for each interpolation. Due to the additional computational cost, it is not possible to achieve realtime rendering performance on current hardware, especially when interpolating between several images at once,

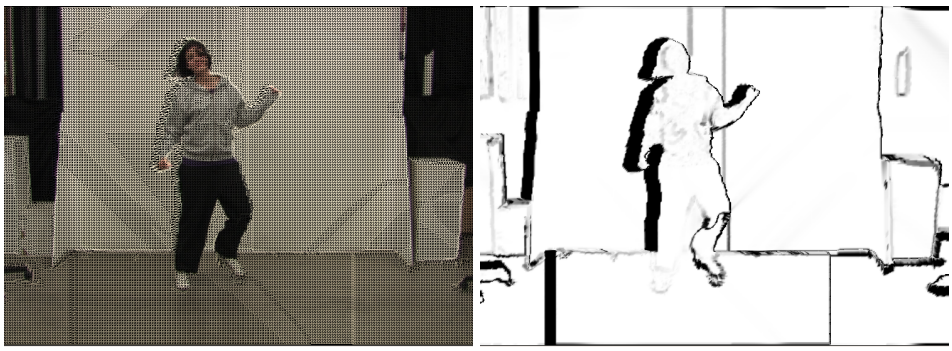

Figure 5.1: *Left: Per-vertex mesh deformation is used to compute the forward warping of the image, where each pixel corresponds to a vertex in the mesh. The depicted mesh is at a coarser resolution for visualization purposes. Right: The connectedness of each pixel that is used during blending to avoid a possibly incorrect influence of missing regions.*

where multiple correspondence fields have to be computed. The results shown in this paper are therefore created using the simpler linear motion interpolation, which is in most cases sufficient and indistinguishable from the per-transformation interpolation.

5.2 Warping with Discontinuities

The first step in creating the in-between images is warping the images. We will use the following notation to denote a warping of an image A by the linearly interpolated deformation field $d_{AB}(\alpha)$ to get the interpolated image $I(\alpha)$

$$I(\alpha) = A \circ d_{AB}(\alpha) \tag{5.2}$$

Unfortunately, for linear motion interpolation a simple remapping is not possible since the inverse motion field cannot be interpolated due to the motion discontinuities. Thus, we resort to forward warping by using a regular planar triangle mesh for the image plane, where each pixel in the image is represented by a vertex in the mesh with appropriate texture coordinates. The vertices are moved with the corresponding motion vector to produce the warped image. Two problems arise with forward warping at motion discontinuities: Fold-overs and missing regions.

Fold-overs occur when two or more pixels in the image end up in the same position after warping. This is, for example, the case when the foreground occludes parts of the background after the warping. While this might be easily solved using the standard depth buffer, we do not know the true depth of a pixel (since we did not need to calculate it for the interpolation). Fortunately, it is sufficient to know the relative depths between the overlapping parts, and one can often use simple

Figure 5.2: *To remove jaggy artifacts at motion discontinuities, a small low-pass filter is applied for feathering.*

heuristics to compute them. Consistent with the perceptual effect known as 'motion parallax', we assume that the faster moving pixel is in the foreground. Should this heuristic fail at a critical fold-over, the user can correct the relative depth on a per superpixel basis.

Missing regions are the opposite of fold-overs. Instead of trying to cut the mesh correctly at motion discontinuities, which is crucial but hard to do robustly, we propose an approach similar in spirit to [MMB97]. We render triangles that span these discontinuities as though they were correct, and measure the reliability of each vertex with the so-called connectedness c_i of each vertex v_i which is computed by

$$c_A = 1 - div(d_{AB})^2. \tag{5.3}$$

The connectedness is used during image blending to get the correct final result. Thus triangles that are stretched during warping have a low connectedness and have less influence on the final result. By using a slightly larger support when computing the divergence, we achieve a smooth transition of the connectedness into the missing regions and avoid artifacts due to slight color changes of corresponding regions when interpolating between different cameras.

5.3 Feathering

At fold-overs, the warped images have jaggy artifacts since the mesh is not rendered with anti-aliasing at the boundaries. Thus, in contrast to naturally observed scenes, pixels at the boundaries are not a mixture of background and foreground but are either foreground or background. Since these artifacts occur only at motion discontinuities, they can be easily marked by using a threshold on the local change in the motion vectors. In a second rendering pass, we then apply a small selective

low-pass filter in only those marked pixels, which simulates the natural mixing of foreground and background at boundaries. This effectively removes the artifacts with a minimal impact on rendering speed and preserves all high-frequency details in the non-discontinuous regions.

5.4 Multiple Image Interpolation

We can describe the interpolation between two images A and B as

$$I(\alpha) = \frac{c_A(1-\alpha) \cdot [A \circ d_{AB}(\alpha)] + c_B(\alpha) \cdot [B \circ d_{BA} \cdot (1-\alpha)]}{c_A(1-\alpha) + c_B(\alpha)} \tag{5.4}$$

where $c_X(\phi)$ is the locally varying influence of each image on the final result which is modulated by the connectedness

$$c_A(\alpha) = c_A \cdot \alpha \tag{5.5}$$

Thus, the (possibly incorrect) influence of pixels with low connectedness on the final result is reduced.

The interpolation is not restricted to two images. Interpolating between multiple images is achieved by iteratively repeating the warping and blending as described in (5.4), where I takes over the role of one of the warped images in the equation. To stay inside the image manifold that is spanned by the images the interpolation factors must sum to one, $\sum_i \alpha_i = 1$. We easily achieve real-time performance on a NVIDIA GeForce 7900GTX.

Chapter 6

User Study

In order to assess the perceptual quality of the proposed algorithm, we ran a psychophysical validation study which had three major goals:
1. to compare the results of the proposed pipeline against standard approaches to image warping
2. to quantify changes in perceptual quality introduced by parameter changes *within* the pipeline
3. to investigate whether there would be a perceptual difference between results on real-world and synthetic image material

6.0.1 Stimuli

Using these criteria, we selected a total of nine different approaches for creating interpolated image sequences. The input to all algorithms consisted of several image sequences depicting a camera rotation around an object. From these sequences we kept every third frame as ”keyframes” and used the algorithms for interpolating the missing two intermediate frames. The following list describes the algorithms in more detail:
original: as the baseline, we compared all algorithms against the original video sequence showing the full, smooth motion
blend: a simple blending algorithm which creates intermediate frames by blending between two consecutive keyframes
opticalflow: by employing a standard optical-flow algorithm [HS81], we established frame-to-frame correspondences which were used to do a simple linear warping between the two keyframes
nooptim: initial solution after the second iteration without optimization of the correspondence field
optim100: the output of the pipeline after the second iteration constraining the correspondence field to be flexible

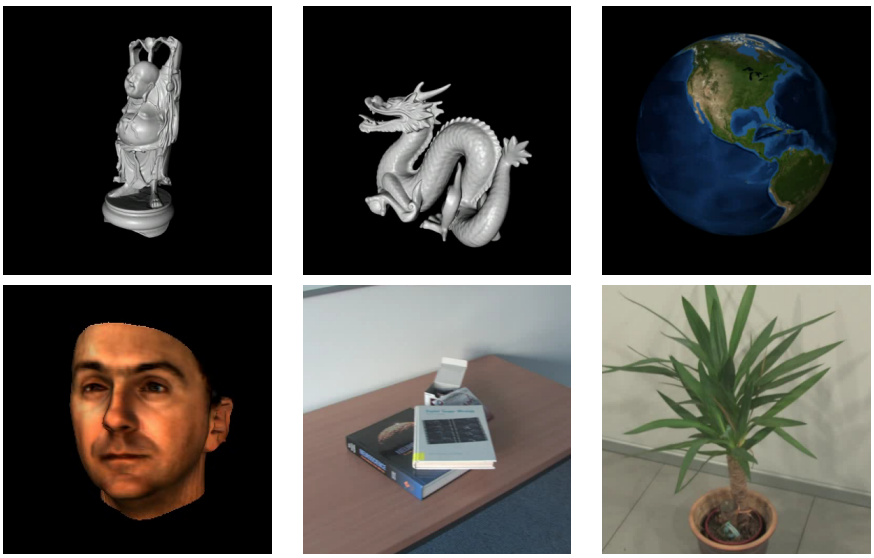

Figure 6.1: *The six different scenes used in the psychophysical validation study. The first four scenes consist of computer-generated 3D objects, whereas the fifth and the sixth scene were recorded indoors with a standard hand-held camera.*

nofeathering: the output of the full pipeline including diffusion but without the feathering at motion discontinuities
firstit: the output of the pipeline after the first iteration with optimization of the correspondence field and subsequent diffusion
full: the output of the full pipeline after the second iteration with optimization of the correspondence field and diffusion
corrected: the output of the full pipeline with a few matches corrected by hand

The first three conditions together with the *full, corrected* conditions address the first goal of comparing different algorithms for interpolation, whereas conditions *firstit, nooptim, optim100, nofeathering* were designed to compare the perceptual quality of different parameter settings.
In order to address our third goal of comparing performance differences of the pipeline on real-world and synthetic images, we used the two different types of scenes shown in Figure 6.1. Four scenes showed computer-generated sequences of a 3D object rotating smoothly around the vertical axis for 180 degrees. The two real-world scenes showed a plant and a table with books which were recorded with a standard, hand-held digital video camera.

6.0.2 Experimental design

Rather than using a standard rating task in which participants would be shown a sequence and be asked to rate its quality, we opted for a more systematic approach.

In the psychophysical study, we used a two-alternative-forced-choice task in which two video sequences were shown successively and participants were asked to indictate which sequence contained more visual artifacts. Such a direct comparison allows for a more fine-grained analysis of the data as rating tasks are often subject to scaling problems [WBF+07]. For each of the 6 different scenes we compared all 9 different interpolation algorithms against each other (only doing AB and AA, not BA comparisons), yielding a total of $6 \cdot (9 \cdot \frac{8}{2} + 9) = 270$ trials.

All scenes were rendered at 500x500 pixels with 25 frames per second and were 3-5 seconds long. Sequences were presented on a black background on a CRT monitor using a pixel resolution of 1024x768 at 75Hz. Participants viewed the stimuli at a distance of roughly 50cm while sitting in a dark room. Each trial consisted of a fixation cross shown for 1 second, followed by the first sequence, a second fixation cross for 0.5 seconds, and the second sequence. After this, the screen was blanked and participants were asked to indicate by a keypress which sequence contained more visual artifacts. Participants were briefed before the experiment that in this case artifacts were defined as "any visual disturbances resulting in non-smooth motion". All participants completed three test trials before the experiment, which were used to get them acquainted with the task. Neither during the test trials nor during the experiment was any feedback given and none of the participants reported any difficulty with doing the task. The whole experiment lasted around 90 minutes. Our test group consisted of 10 participants who did *not* have any graphics-related background.

6.0.3 Analysis

For the first analysis, we determined a perceptual quality score by counting how many times a particular algorithm was chosen as producing less visual artifacts. The normalized scores are shown in Figure 6.2 for all nine algorithms. The following analysis addresses our first two experimental questions, by interpreting the results for each algorithm (all statistical tests were run as one-tailed t-tests corrected for multiple comparisons):

original: The original sequences are rated as having the best perceptual quality (all $p<0.01$).
blend: Despite the technical simplicity of this condition, the quality score is still reasonably high. Whereas this might be surprising at first glance, the perceptual impression of the resulting motion is that of a jerky, but very consistent motion.
opticalflow: This condition was rated as having the worst quality (all $p<0.01$). Even though we used a standard implementation of the Horn-Schunck approach [HS81], the algorithm failed to find correct correspondences in most cases resulting in a large number of noticable artifacts at stimulus edges. This violation of object contour stability destroys the perceptual quality of the sequences completely.
nooptim: Of all the approaches based on the proposed pipeline, this was the worst

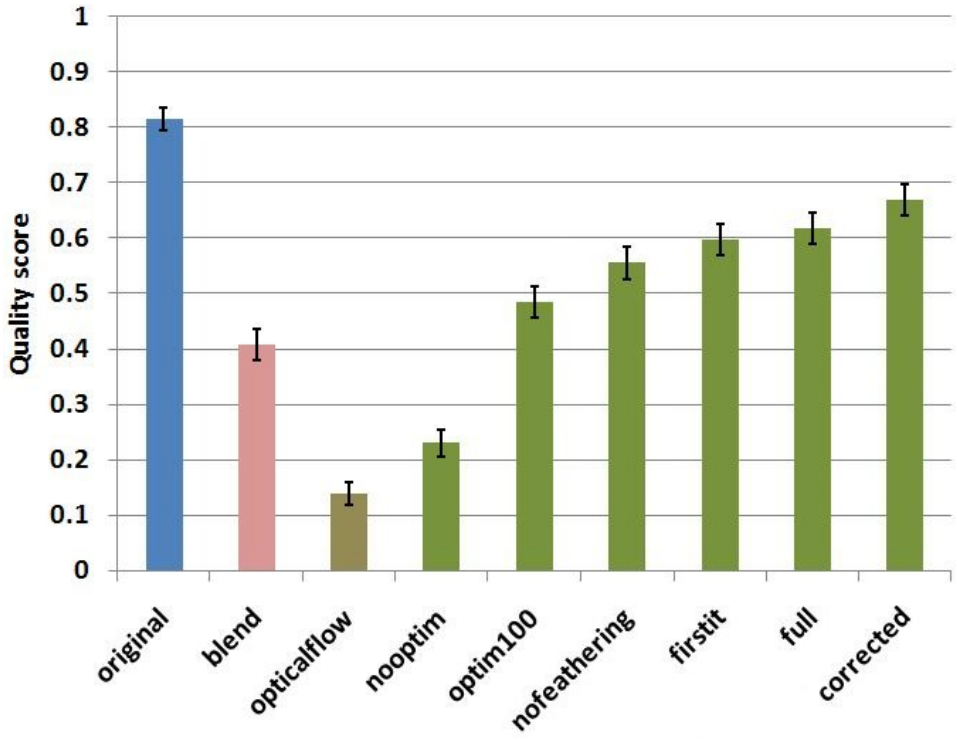

Figure 6.2: *Perceptual quality scores for nine different test conditions (image interpolation schemes).*

condition (all $p<0.01$). In several cases, this algorithm resulted in sharp spikes and discontinuities in the correspondence field which occurred in otherwise homogenous image regions. Given that our visual system is highly sensitive to sudden changes in image intensities (even if these take up only a very small proportion of the image), these results are not surprising.

optim100: Compared to the *nooptim* condition, the increase in perceptual quality due to the optimization of the correspondence field is dramatic demonstrating the importance of producing locally consistent motion for perceptual fidelity.

nofeathering: According to our expectations the feathering of edges for our scenes has a perceptually noticeable positive effect.

firstit: The results show that already after the first iteration, the proposed pipeline produces perceptually pleasing motion as reflected in the high quality scores. Compared to the *optim100* condition, there is another significant increase in perceptual quality. This increase is due to the diffusion post-processing, which enforces global consistency of the results.

full: The perceptual quality shows that having one iteration seems to be perceptually as good as having two iterations.

corrected: Not surprisingly, of all the approaches based on the proposed pipeline this condition fared best (all $p<0.05$). The difference between this and the full con-

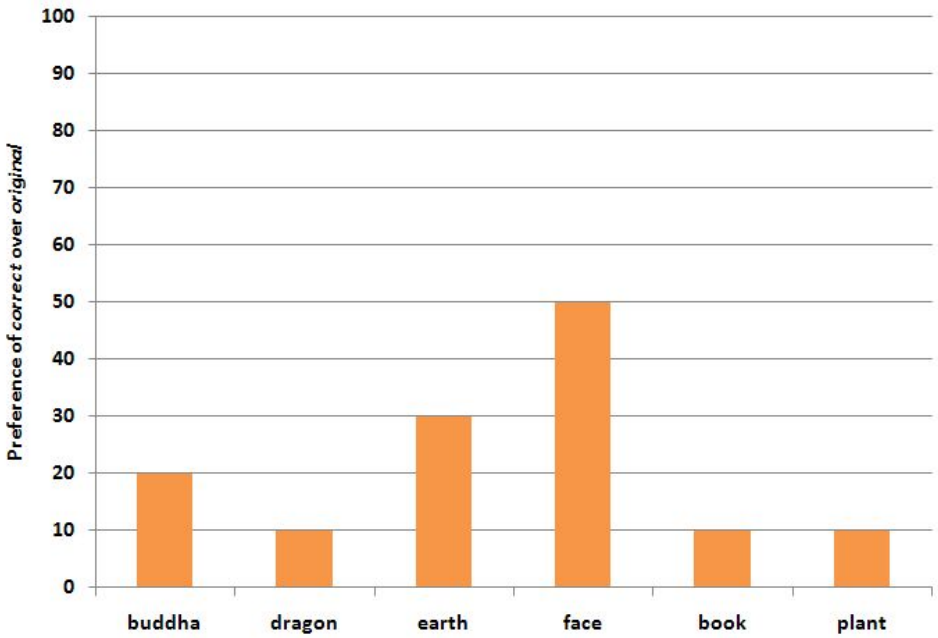

Figure 6.3: *Preference of corrected over original condition, broken down by test scene. 50 percent denotes that both conditions are of equal perceived quality.*

dition is small but significant showing that a small amount of human intervention can improve the results further.

In order to address the third experimental question of quality differences between real-world and synthetic scenes we compared how many times participants chose the *corrected* over the *original* condition. As Figure 6.3 shows, for both real-world scenes, only one response was given in favor of the corrected scene, whereas for the face sequence it seems that participants could not decide which of the two conditions was better, as preference was at 50%.

Taken together, these results have shown that the proposed pipeline already produces perceptually plausible, high-quality interpolations. Whereas there is still some room for improvement - especially for identifying invalid correspondences and improving robustness against single outliers - the quality of the sequences is surprisingly good given that no prior knowledge about camera calibrations, scene layout, or object identity was used. Additionally, the results confirm and extend the perceptual approach to computer graphics - that our visual system has evolved to deal with natural *image statistics* (things tend to move smoothly; objects have well-defined, stable boundaries, etc.) rather than to explicitly and accurately reconstruct the 3D world from visual input (simple warping can be enough).

Chapter 7

Results

Our perception-based image interpolation approach can be applied to any set of sufficiently similar real-world images. While we can only show here a small number of the visual effects and applications made possible, we hope to convincingly demonstrate the versatility and usefulness of perception-based image interpolation.

Space-Time Interpolation In image- and video-based rendering, the viewpoint can be navigated freely to interactively regard the static or dynamic scene from any vantage point. Likewise, when interpolating between images taken from different positions, the impression of authentic viewpoint motion can be achieved. In addition, image interpolation can be applied to images taken at different moments in time, allowing to create scene views for intermediate time points as well as in-between camera positions. Given our pipeline, dynamic scene recordings with conventional, unsynchronized, and uncalibrated video cameras suffice to continuously navigate the virtual viewpoint in space and time. Examples are shown in the accompanying video, interpolating in-between views from up to four different camera frames.

The fundamental limitation of image interpolation is, of course, that the viewpoint always lies between camera recording positions, i.e., in the (triangulated) surface spanned by the camera positions during acquisition. Other image-based rendering techniques are able to render the scene from any arbitrary viewpoint. On the other hand, image information is recorded only at camera positions: moving much closer towards the object results in blurred rendering results, while moving away from the object is almost equivalent to rendering the object to smaller scale.

Another point to note is that by ignoring the epipolar constraint, perception-based image interpolation between different cameras runs the danger of introducing visible distortions if camera positions are spaced too far apart [SD96]. For our recordings, we used Canon HDV 1080p camcorders that had a horizontal field of view

of$\approx 50^\circ$ and were spaced apart by $\approx 15^\circ$. With these acquisition parameters, no distortion is apparent.

Global Visual Effects Besides virtual viepwoint navigation, our perception-based interpolation approach is easily extended to create any number of global visual effects from multi-camera recordings. Following the terminology introduced by Taylor and MacLeod [TM08], perception-based space-time image interpolation can readily be used to produce such visual effects as frozen moment, live action, stop start, slow motion, time and space ramp. A few examples are shown in the accompanying video. Additionally, time blur, space blur, long exposure, and multiple exposure shots are easily generated by compositing several perceptually interpolated images. Figure 7.1(a,b) depicts some examples. Going beyond what is possible with capturing visual effects directly [TM08], our approach also allows to add arbitrary vector fields to the correspondence field. We can, for example, compensate for the motion of some scene region, keeping the region at the same position of the image plane. An example for a locally motion-compensated, space blurred image is shown in Figure 7.1(c).

Movie Restauration As a third application for perception-based image interpolation, we consider the problem of restoring and temporally augmenting historic movies. For technical reasons, movies were recorded at inadequately low frame rates up into the 1920ies. Over the decades, the celluloid has aged and today shows scratches and holes. Despite the mediocre quality of the input image, our perception-based interpolation algorithm is able to estimate convincing correspondence fields that enable filling in any damaged regions from previous and future frames, Figure 7.2. By adding temporally interpolated frames, we are also able to achieve modern movie frame rates.

Key frame | Temporal motion blur effect | Key frame

(a)

Key frame | Progressive motion distortion effect | Key frame

(b)

Key frame | Local motion compensation effect | Key frame

(c)

Figure 7.1: *Different visual effects created using perception-based image interpolation.*

(a) Key frame (b) Corrupted frame (c) Key frame (d) Reconstructed frame

Figure 7.2: *Restauration of a corrupted frame.*

Chapter 8

Discussion and Conclusions

In this paper, we have presented a perception-based image interpolation approach. By taking visual motion perception into consideration, we are able to robustly estimate dense correspondence fields between images such that visually convincing interpolation results are obtained. In contrast to other interpolation techniques, our approach is geared towards synthesizing perceptually plausible transitions instead of enforcing physical correctness. Our contributions enable smooth, convincing interpolation across space and time without the need for time-consuming camera calibration, error-prone geometry reconstruction, or an expensive synchronized multi-video acquisition system. Possible applications include image-based rendering, visual effect production, and movie restauration.

The proposed method to estimate the perceptual correspondence field currently considers only pairs of images. Thanks to the perception-centered approach, mismatches between correspondence fields do not lead to visible artifacts. We nevertheless intend to look at global methods to increase coherence of the correspondence field across space and time. This will allow correctly interpolating the motion of scene objects that become completely occluded from one frame to the next.

Bibliography

[ACOL00] Marc Alexa, Daniel Cohen-Or, and David Levin. As-rigid-as-possible shape interpolation. In *Proc. ACM Conference on Computer Graphics (SIGGRAPH'00)*, New Orleans, pages 157–164, 2000.

[BBM+01] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen. Unstructured lumigraph rendering. In *Proc. ACM Conference on Computer Graphics (SIGGRAPH'01)*, Los Angeles, pages 425–432. ACM, 2001.

[Ber92] Dimitri Bertsekas. Auction algorithms for network flow problems: A tutorial introduction. *Computational Optimization and Applications*, 1:7–66, 1992.

[BFB94] J. Barron, D. Fleet, and S. Beauchemin. Performance of Optical Flow Techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[BM04] S. Baker and I. Matthews. Lucas-Kanade 20 Years On: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[BMP01] Serge Belongie, Jitendra Malik, and Jan Puzicha. Matching Shapes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 454 – 461, 2001.

[BN92] Thaddeus Beier and Shawn Neely. Feature-based image metamorphosis. In *Proc. ACM Conference on Computer Graphics (SIGGRAPH'92)*, Chicago, pages 35–42. ACM, 1992.

[Can86] John Canny. A Computational Approach To Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–714, 1986.

[CTMS03] J. Carranza, C. Theobalt, M. Magnor, and H. P. Seidel. Free-viewpoint video of human actors. In *Proc. ACM Conference on Computer Graphics (SIGGRAPH'03)*, San Diego, pages 569–577. ACM, 2003.

[CW93] S. Chen and L. Williams. View interpolation for image synthesis. In *Proc. ACM Conference on Computer Graphics (SIGGRAPH'93)*, Anaheim, pages 279–288. ACM, 1993.

[DBY98] P. Debevec, G. Borshukov, and Y. Yu. Efficient view-dependent image-based rendering with projective texture-mapping. In *Proc. Eurographics Rendering Workshop (EGRW'98)*, pages 105–116, 1998.

[FH04] Pedro Felzenszwalb and Daniel Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59:167–181, 2004.

[GGSC96] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen. The Lumigraph. In *Proc. ACM Conference on Computer Graphics (SIGGRAPH'96)*, New Orleans, pages 43–54. ACM, 1996.

[Gib55] J.J. Gibson. *The Perception of the Visual World.* Cambridge UP, 1955.

[GP00] M. Giese and T. Poggio. Morphable models for the analysis and synthesis of complex motion patterns. *International Journal of Computer Vision*, 38:59–73, 2000.

[GP03] M. Giese and T. Poggio. Neural mechanisms for the recognition of biological movements. *Nature Reviews – Neuroscience*, 4:179–192, March 2003.

[Gra65] C. Graham. *Vision and Visual Perception*, chapter Perception of movement. New York: Wiley, 1965.

[HBD+99] D.J. Heeger, G.M. Boynton, J.B. Demb, E. Seidemann, and W.T. Newsome. Motion opponency in visual cortex. *J. Neurosci.*, 19:7162–7174, Aug 1999.

[HS81] B. Horn and B. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17:185–203, 1981.

[HZ00] R. Hartley and H. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[IMG00] A. Isaksen, L. McMillan, and S. Gortler. Dynamically reparameterized light fields. In *Proc. ACM Conference on Computer Graphics (SIGGRAPH'00)*, New Orleans, pages 297–306. ACM, 2000.

[LH96] M. Levoy and P. Hanrahan. Light field rendering. In *Proc. ACM Conference on Computer Graphics (SIGGRAPH'96)*, New Orleans, pages 31–42. ACM, 1996.

[LK81] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Seventh International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[Low04] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[LTF+05] Ce Liu, Antonio Torralba, William T. Freeman, Frédo Durand, and Edward H. Adelson. Motion magnification. In *Proc. ACM Conference on Computer Graphics (SIGGRAPH'05)*, San Diego, pages 519–526, 2005.

[Mar82] David Marr. *Vision.* Freeman, 1982.

[MB95] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. *Proc. ACM Conference on Computer Graphics (SIGGRAPH'95)*, Los Angeles, pages 39–46, August 1995.

[MBR+00] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan. Image-based visual hulls. In *Proc. ACM Conference on Computer Graphics (SIGGRAPH'00)*, New Orleans, pages 369–374. ACM, 2000.

[MH80] David Marr and Ellen Hildreth. Theory of Edge Detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 207:187–217, 1980.

[MMB97] William Mark, Leonard McMillan, and Gary Bishop. Post-Rendering 3D Warping. In *Proceedings of the Symposium on Interactive 3D Graphics*, pages 7–16, 1997.

[MP04] W. Matusik and H. Pfister. 3D TV: A scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes. In *Proc. ACM Conference on Computer Graphics (SIGGRAPH'04)*, Los Angeles, pages 814–824. ACM, 2004.

[MTAS01] Karol Myszkowski, Takehiro Tawara, Hiroyuki Akamine, and Hans-Peter Seidel. Perception-guided global illumination solution for animation rendering. In *Proc. ACM Conference on Computer Graphics (SIGGRAPH'01)*, Los Angeles, pages 221–230, New York, NY, USA, 2001. ACM.

[OHM+04] Carol O'Sullivan, S. Howlett., R. Mcdonnell, Y.Morvan, and K K. O'Conor. Perceptually adaptive graphics. *In Eurographics 04, State-of-the-art-Report 6*, 2004.

[PM90] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[QA97] N. Qian and R.A. Andersen. A physiological model for motion-stereo integration and a unified explanation of Pulfrich-like phenomena. *Vision Res.*, 37:1683–1698, Jun 1997.

[Rei61] W. Reichardt. Autocorrelation, a principle for the evaluation of sensory information by the central nervous system. In W. Rosenblith, editor, *Sensory communication*, page 303317. New York: MIT Press-Willey, 1961.

[RFWB07] Ganesh Ramanarayanan, James Ferwerda, Bruce Walter, and Kavita Bala. Visual equivalence: Towards a new standard for image fidelity. In *Proc. ACM Conference on Computer Graphics (SIGGRAPH'07)*, San Diego, pages 654–663, 2007.

[RM03] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 10– 17, 2003.

[RT99] Mark Ruzon and Carlo Tomasi. Color Edge Detection with the Compass Operator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 160–166, 1999.

[SD96] Steven M. Seitz and Charles R. Dyer. View morphing. In *Proc. ACM Conference on Computer Graphics (SIGGRAPH'96)*, New Orleans, pages 21–30. ACM, 1996.

[SMW06] Scott Schaefer, Travis McPhail, and Joe Warren. Image deformation using moving least squares. In *Proc. ACM Conference on Computer Graphics (SIGGRAPH'06)*, Boston, pages 533–540. ACM, 2006.

[SSS06] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *Proc. ACM Conference on Computer Graphics (SIGGRAPH'06)*, Boston, pages 835–846. ACM, 2006.

[TM08] D. Taylor and H. McLeod. Digital air - techniques, 2008. http://www.digitalair.com/techniques/index.html.

[VBK05] S. Vedula, S. Baker, and T. Kanade. Image based spatio-temporal modeling and view interpolation of dynamic events. *ACM Transactions on Graphics*, 24(2):240–261, April 2005.

[VBR+05] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(3):475–480, mar 2005.

[VLD07] Peter Vangorp, Jurgen Laurijssen, and Philip Dutr. The influence of shape on the perception of material reflectance. In *Proc. ACM Conference on Computer Graphics (SIGGRAPH'07)*, San Diego, pages 1–9, 2007.

[WAA+00] D. Wood, D. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. Salesin, and W. Stuetzle. Surface light fields for 3D photography. In *Proc. ACM Conference on Computer Graphics (SIGGRAPH'00)*, New Orleans, pages 287–296. ACM, 2000.

[Wal35] H. Wallach. Ueber visuell wahrgenommene bewegungsrichtung. *Psychologische Forschung*, 20:325–380, 1935.

[WBF+07] C. Wallraven, H. H. Bülthoff, J. Fischer, D. W. Cunningham, and D. Bartz. The evaluation of real-world and computer-generated stylized facial expressions. *ACM Transactions on Applied Perception*, 4(3):1–24, 2007.

[Wer38] Max Wertheimer. Laws of organization in perceptual forms. In W.D. Ellis, editor, *A Source Book of Gestalt Psychology*, pages 71–88. Kegan Paul, Trench, Trubner & Co. Ltd., 1938.

[Wol98] George Wolberg. Image morphing: A survey. *Visual Computer*, 14:360–372, 1998.

[Wol06] M. Wolf. Space, time, frame, cinema: Exploring the possibilities of spatiotemporal effects. *New Review of Film and Television Studies*, pages 369–374, December 2006. www.digitalair.com/techniques/STFC.pdf.

[ZKU+04] C. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. In *Proc. ACM Conference on Computer Graphics (SIGGRAPH'04)*, Los Angeles, pages 600–608. ACM, 2004.